

Z-code Text Adventure Collection #1 for MEGA65

Many of us played and loved the text adventures (AKA interactive fiction) produced by Infocom in the 1980's. They were rich in story and puzzles, and contained some excellent writing. In the years since Infocom's demise in 1989, there have been a lot of good games produced using Z-code - the game format that Infocom was using. Many of these games run fine on the MEGA65, using a Z-code interpreter for the MEGA65. There are currently two Z-code interpreters for the MEGA65 - Bit Shifter's Z Engine and Ozmoos. This collection uses Ozmoos.

I decided to create a collection of some great Z-code games for the MEGA65. All in all, this collection holds ten games, which were initially released in 1993-2021.

If you want more of a background on Infocom and the game format they used, you should read the chapter about The Zork Machine at the end of this document. There is also a chapter about building other Z-code games for play on MEGA65 and, finally, a chapter about writing your own games.

If you are new to text adventures, you may want to start by reading a short guide on how to play at <https://www.microheaven.com/ifguide/step3.html> or you could start by playing The Dreamhold, which was written to guide new players.

I created this documentation as a PDF, so that you could easily print it out and keep it nearby as you're playing the games on your MEGA65.

Enjoy!

Fredrik Ramsberg, July 2022

Email: fredrik.ramsberg AT gmail.com

The Games	3
Adventure	3
Ad Verbum	3
Aisle	4
Curses	4
Djinn On the Rocks	5
The Dreamhold	5
The Fire Tower	6
Make It Good	6
Pytho's Mask	7
Theatre	7
Technical Information	8
The Zork Machine - past and present	8
Building Z-code games for play on MEGA65	10
Writing your own text adventures	11

The Games

Adventure

Disk image: ADVENT.D81

Genre: Fantasy

Year: 2015

Release / Serial No: 1 / 151001

Authors: William Crowther and Donald Woods

Programmer: Jesse McGrew

Link: <http://ifdb.tads.org/viewgame?id=fft6pu91j85y4acv>

This is based on the 350-point version of Adventure, sometimes called Colossal Cave Adventure. The original game was released by Will Crowther in 1976 and expanded by Don Woods in 1977. This was the game that started the text adventure genre, and the inspiration behind Zork. This version was written in ZIL, the language used by Infocom to program Zork and their other Z-machine games.

Ad Verbum

Disk image: ADVERBUM.D81

Genre: Humor / Wordplay

Year: 2000

Release / Serial No: 11 / 060905

Author: Nick Montfort

Link: <http://ifdb.tads.org/viewgame?id=xi4s5ne9m6w821xd>

With the cantankerous Wizard of Wordplay evicted from his mansion, the worthless plot can now be redeveloped. The city regulations declare, however, that the rip-down job can't proceed until all the items within had been removed.

As an adventurer hired by the demolitions contractor to kleptomaniacally clear out this mansion, you must engage in wordplay in order to gather all the items inside. It is not necessary to think of puns, cliches, or homonyms, however, as has been the case with previous logological interactive fiction. The puzzles in Ad Verbum are of a different -- and perhaps even unique -- nature.

[--blurb from Competition Aught-Zero]

Aisle

Disk image: AISLE.D81

Genre: Slice of Life

Year: 1999

Release / Serial No: 1 (revision 3) / 990528

Author: Sam Barlow

Link: <http://ifdb.tads.org/viewgame?id=j49crlvd62mhwuzu>

Late Thursday night. You've had a hard day and the last thing you need is this: shopping. Luckily, the place is pretty empty and you're progressing rapidly. On to the next aisle...

Aisle started out as a game which would not need the usual meta-verbs... i.e. a game with only one turn. The initial idea was: How do I make a game with only one turn interesting? Give it lots of endings--in fact there are many 'endings' and (hopefully) every sensible action results in an 'ending'. There is no winning action. There is however more going on than just this and the more endings you see the more things should become clear.

[--blurb from The Z-Files Catalogue]

Sam Barlow has since gone on to design some of the Silent Hill games, and has achieved great success on the indie game scene with Her Story.

Curses

Disk image: CURSES.D81

Genre: Haunted House / Historial / Travel / Time Travel

Year: 1993

Release / Serial No: 16 / 951024

Author: Graham Nelson

Link: <http://ifdb.tads.org/viewgame?id=plvzam05bmz3enh8>

As "Curses" opens, you're hunting about in the attic of your family home, looking for a tatty old map of Paris (you're going on holiday tomorrow) and generally trying to avoid all the packing. Aunt Jemima is potting daisies and sulking; the attics are full of endless distractions and secrets; Greek myths, horoscopes, sixth-century politics, a less than altogether helpful demon, a mysterious bomb plot, photography, ritual, poetry and a dream or two all get in your way; and somehow you keep being reminded of your family through the ages, and all its Curses... ...could it be that even you are Cursed?

[--blurb from The Z-Files Catalogue]

Graham Nelson created the Inform programming language, which he then used to develop his first large interactive fiction game, Curses. This most recent release of Curses was made using Inform 5. Graham Nelson has gone on to write several highly acclaimed games.

Djinn On the Rocks

Disk image: DJINNOTR.D81

Genre: Fantasy

Year: 2021

Release / Serial No: 2 / 210519

Author: Joshua Wilson

Link: <https://ifdb.org/viewgame?id=j9c3fyfv7ub5m7pa>

James Merl III is a thrice-cursed pain in your magically imprisoned butt. What are you going to do about it?

A short comic fantasy adventure game featuring mild but mischievous reality bending magic powers in the form of parser-based interactive fiction created for and submitted to the PunyJam#1 competition.

[--blurb from IFDB]

This game was developed with PunyInform, in a 3-week game jam.

The Dreamhold

Disk image: DREAMHLD.D81

Genre: Fantasy

Year: 2004

Release / Serial No: 5 / 041231

Author: Andrew Plotkin

Link: <https://ifdb.org/viewgame?id=3myqnrs64nbtwdaz>

The Dreamhold is designed for people who have never played IF before. It introduces the common commands and mindset of text adventures, one step at a time. There's an extensive help system describing standard IF commands, as well as dynamic hints which pop up whenever you seem to be stuck.

I've tried to create a game which rewards many species of adventurer: the inexperienced newcomer, the puzzle-hurdler, the casual tourist, the meticulous explorer, the wild experimenter, the seeker after nuances and implications.

[--blurb from IFDB]

Andrew Plotkin has written a wide array of interactive fiction games, and won many awards. He has also done professional game development, including the highly polished IF game Hadean Lands which can be found on Steam.

The Fire Tower

Disk image: FIRETOWR.D81

Genre: Slice of Life / Travel

Year: 2004

Release / Serial No: 1 / 040528

Author: Jacqueline A. Lott

Link: <https://ifdb.org/viewgame?id=fcm1ikz9ttr6i99a>

This is a game about hiking and exploring the landscape. There are no puzzles, but you can interact with a lot of your surroundings. If you enjoy the experience, you win.

Make It Good

Disk image: MAKEGOOD.D81

Genre: Mystery

Year: 2009

Release / Serial No: 15 / 091227

Author: Jon Ingold

Link: <https://ifdb.org/viewgame?id=jdrbw1htq4ah8q57>

The call comes through. Of all the dicks; you get the call, sitting in the front seat of your car, hands shaking on the steering wheel. An urgent call; but all you were thinking of was the bottle in the liquor store and so that's where you went first.

Now you're pulled up outside the house. The rear mirror's showing two steely eyes. You adjust your hat, stiffen up your collar and grab your badge off the dash. Here goes. You've one last chance so...

MAKE IT GOOD

An Interactive Detective Story, by Jon Ingold

[--blurb from IFDB]

Jon Ingold has moved on from his hobbyist roots in interactive fiction to co-founding Inkle Studios where he is also the narrative director. He has been involved as a writer in critically acclaimed games such as 80 Days, Heaven's Vault and Overboard!

Pytho's Mask

Disk image:PYTHO.D81

Genre: Romance

Year: 2001

Release / Serial No: 3 / 020223

Author: Emily Short

Link: <https://ifdb.org/viewgame?id=spsnevez1br6f2z3v>

On the Night of the Comet, the usual astrological bonds do not hold, and the order of the universe is threatened. It is a time made for rebels and usurpers, and all who would claim the kingdom for themselves.

You are a member of the Order of the Phoenix, a protector of the hierarchy and the kingdom itself. It is your duty to attend the royal ball, watch for dangers... and do whatever needs to be done.

Choose your enemies carefully, and guard your heart.

[--blurb from IFDB]

Emily Short has written many highly acclaimed pieces of interactive fiction, and moved on to a professional career in the game industry. She is currently the creative director at Failbetter Games.

Theatre

Disk image: THEATRE.D81

Genre: Horror

Year: 1995

Release / Serial No: 2 / 951203

Author: Brendon Wyber

Link: <https://ifdb.org/viewgame?id=bv8of8y9xeo7307g>

Your job as a real estate agent brings you into contact with many old buildings, but none are quite like the old theatre that has stood deserted for almost thirty years. After visiting it with some prospective buyers, you discover that you have left your pager behind. You quickly stop off there, on your way out for the evening, to pick it up.

[--blurb from IFDB]

Technical Information

Here you can find a bit more information about Infocom, the game format they developed, how you can find more Z-code games and make them playable on the MEGA65, and what you need to write your own text adventure.

The Zork Machine - past and present

In 1977, a few guys at Massachusetts Institute of Technology played Adventure, which is widely considered to be the first text adventure. They decided to write their own adventure game, first called Zork, then Dungeon, then Zork again. In 1979, they started a company called Infocom, to make business software. They needed a source of income as they were starting up, and they thought Zork could provide that, if they could only get it to run on the micro computers of the time. The game was about 1 MB in size, and the computers they needed to run it on typically had 32 KB of memory. Also, the game was written in Fortran, which wasn't available for micro computers. On top of all this, there were new computer models coming out all the time, and they wanted the game to run on as many different platforms as possible.

To solve this equation, they designed a virtual machine, an imaginary computer never meant to be built, but to be emulated. The machine was called the Z-Machine (Zork Machine), and it was designed only for running text adventures. They then created their own programming language called ZIL (Zork Implementation Language), heavily inspired by MDL, the language they wrote the first version of Zork in. They wrote a compiler called Zilch, which compiled ZIL programs to Z-code, programs ready to be run on the Z-machine. They then wrote a separate ZIP (Z-machine Interpreter) to emulate the Z-machine on each platform they wanted to support. The end result was that they could, after the hard work of writing all this software was done, release their text adventures for many different platforms without rewriting any of the code.

While the Z-machine was heavily optimized to make text adventures compact in size, Zork was still too big. They split up the game into three parts, added some pieces and removed others, to make three different games, each one of them playable on a micro computer with a disk drive attached. An entire game would fit on a disk, and the computer would use virtual memory, copying the parts of the game that it needed at the moment from disk to memory. In this way, games of up to 128 KB in size could be played on a computer with only 32 KB of memory. Zork I was released in 1980, and became a commercial success. Infocom kept on making text adventures for about ten years, producing some of the finest games of the eighties.

Fast forward to the nineties. Infocom had been bought by Activision and eventually shut down. Some fans gathered together over the now emerging Internet, and started figuring out just exactly how the Z-Machine worked, by reading the machine code of the interpreters used (ZIP:s). They called themselves The Infocom Taskforce. A guy in England called

Graham Nelson saw what was happening and started to write his own programming language, compiler and text adventure programming library, based on the findings of the Infocom Task Force. He called his system Inform. While he was developing it, we also wrote his own game in Inform, called Curses. When Inform hit version 5, it had matured enough for a much wider audience to use it. Graham made Inform available for free, along with Curses.

Since Curses was using the same virtual machine as Infocom had used, it could also be played with the same ZIP:s that Infocom had produced, but soon enough other people started to write new Z-machine interpreters, for modern computers. Graham also created two new versions of the Z-machine, called version 7 and 8 (Infocom created versions 1-6), which allowed for bigger games. A version 8 game can be up to 512 KB in size, which allows for truly huge text adventures. It has been estimated that Infocom's first two trilogies (Zork 1, 2, 3 + Enchanter, Sorcerer and Spellbreaker) could be combined into a single game without hitting this limit.

Inform 5 was used to produce a dozen games or so. Then came Inform 6, which has been used to write hundreds of games. After that came Inform 7, which is very different from the earlier versions. It has been used to create hundreds of games as well. To allow for even larger games, and to make it easier to incorporate graphics and sound into text adventures, a new virtual machine, Glulx, was designed by Andrew Plotkin. Inform 7, as well as the more recent versions of Inform 6, can produce both Z-code and Glulx games.

Apart from Inform 6 and 7, there are now three somewhat established ways of producing games in Z-code format:

PunyInform: This is a lightweight library for the Inform 6 programming language, which is useful for anyone who wants to produce games that run fine on 8-bit computers as well.

ZIL: There is a compiler called ZILF to compile a game written in ZIL to a Z-code file. This can be used to compile the original source code of the Infocom games or produce new games.

Dialog: A language based on Prolog, which can be compiled to Z-code.

Building Z-code games for play on MEGA65

There are currently two Z-code interpreters for the MEGA65: Bit Shifter's Z Engine and Ozmoos. I am one of the authors of Ozmoos, and have chosen to use it for this collection.

Bit Shifter's Z Engine and instructions on using it can be found at <https://files.mega65.org?id=de8e640d-3649-4ecb-b6b9-d54534834277>

Ozmoos and instructions on using it can be found at <https://files.mega65.org?pr=96c0bde1-34ad-4a97-a0e3-27029c41d90f>

For Ozmoos, you go to a website, upload a Z-code file, specify your build options like colour scheme etc, and press a button. This produces a disk image which you can then download.

All games produced by Infocom except the four titles with graphics (Zork Zero, Arthur, Shogun and Journey) should work fine on the MEGA65.

As for Z-code games produced post-Infocom, you can generally expect all games that weren't built using Inform 7 to work well. Some lightweight Inform 7 games may also work reasonably well, but most Inform 7 games are just too slow. Inform 7 typically produces code that is very demanding for the computer, and a 40 MHz 8-bit computer isn't up to the task.

A few modern games show some Unicode characters which can't be displayed on a MEGA65. These characters will just be omitted in output when playing on a MEGA65. If you want to play a game in German, French, Spanish, Italian, Danish or Swedish, Ozmoos has you covered though, as it can use a custom font which replaces some graphic characters with the accented characters you need.

Writing your own text adventures

You can of course write your own text adventures as well. Some popular tools include Inform 7, Twine, ChoiceScript, TADS 3 and Adrift.

If you want to produce Z-code games, and have them playable on retro computers, these are the obvious options today:

- ZIL: Use the Lisp-like language used by Infocom. The compiler and some pointers can be found at <https://foss.heptapod.net/zilf/zilf>
- Dialog: A language inspired by Prolog. It's quite advanced and powerful, but the games produced are a bit too heavy to run on your typical 8-bit machine. 16-bit machines and the MEGA65 are better suited to run these games. Find out more at <https://www.linusakesson.net/dialog/>
- Inform 6 with its standard library: Similar to C in syntax. Games produced are a bit too heavy for most 8-bit machines, but typically run fine on MEGA65 and 16-bit machines. There is a (no longer maintained) site at <https://www.inform-fiction.org/>
- Inform 6 with the PunyInform library: With a much more lightweight library, you can produce games that run fine on all sorts of 8-bit computers for which there are Z-code interpreters (~30 different platforms). Of course, the games will also run fine on newer computers. The homepage is located at <https://github.com/johanberntsson/PunyInform>

I am a co-author of PunyInform and perhaps not surprisingly, I think PunyInform is the easiest option for writing Z-code games for retro computers. While the library is designed to be much smaller and faster than the standard library, it has most of the functionality of the standard library, plus some that the standard library lacks.

No matter which system you choose, or even before you choose a system, I recommend the friendly forum at <https://intfiction.org/> as a place to get support, encouragement, feedback, playtesters, and all sorts of useful pointers.